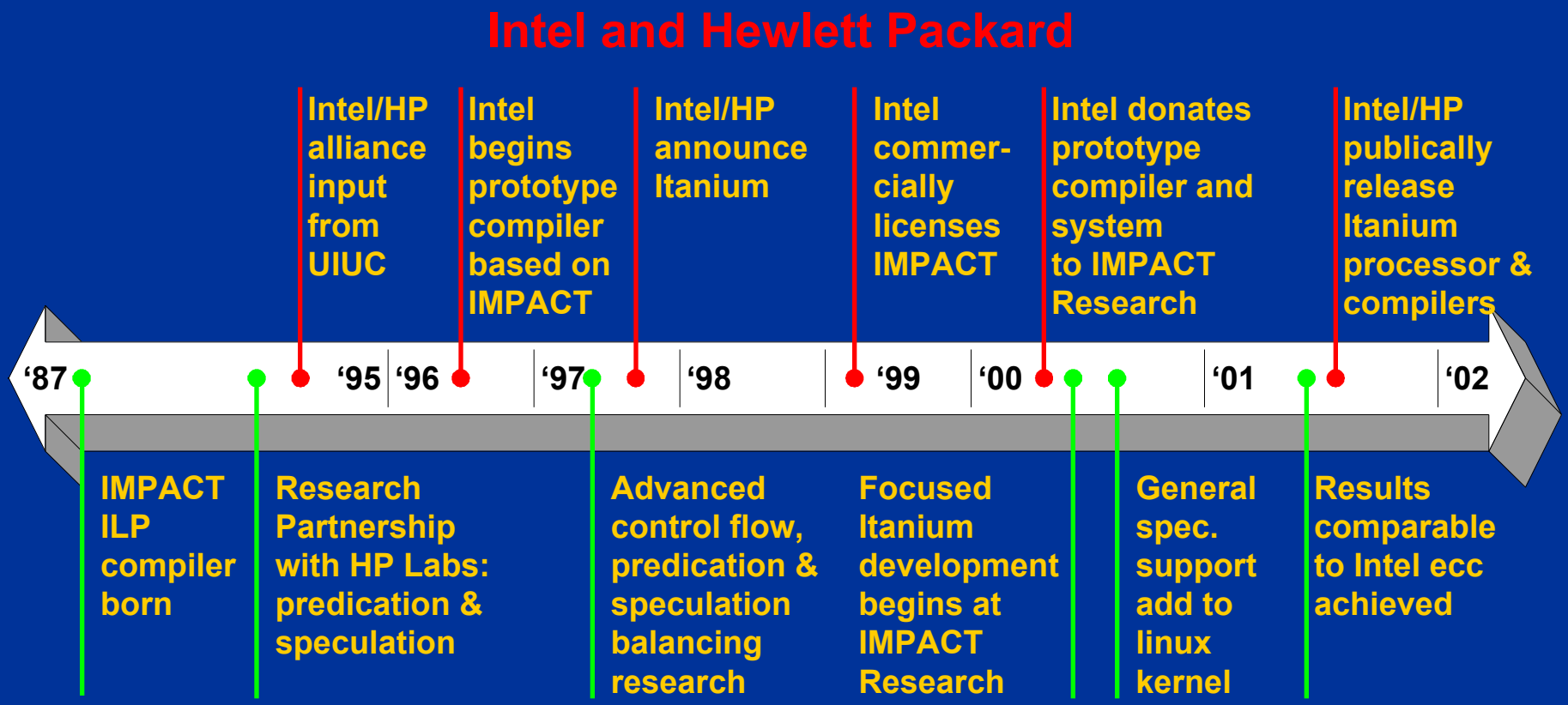
Itanium Performance Insights from the IMPACT Compiler

J. Sias, M. Merten, E. Nystrom, R. Barnes,
C. Shannon, J. Matarazzo, S. Ryoo,
J. Olivier, W. Hwu

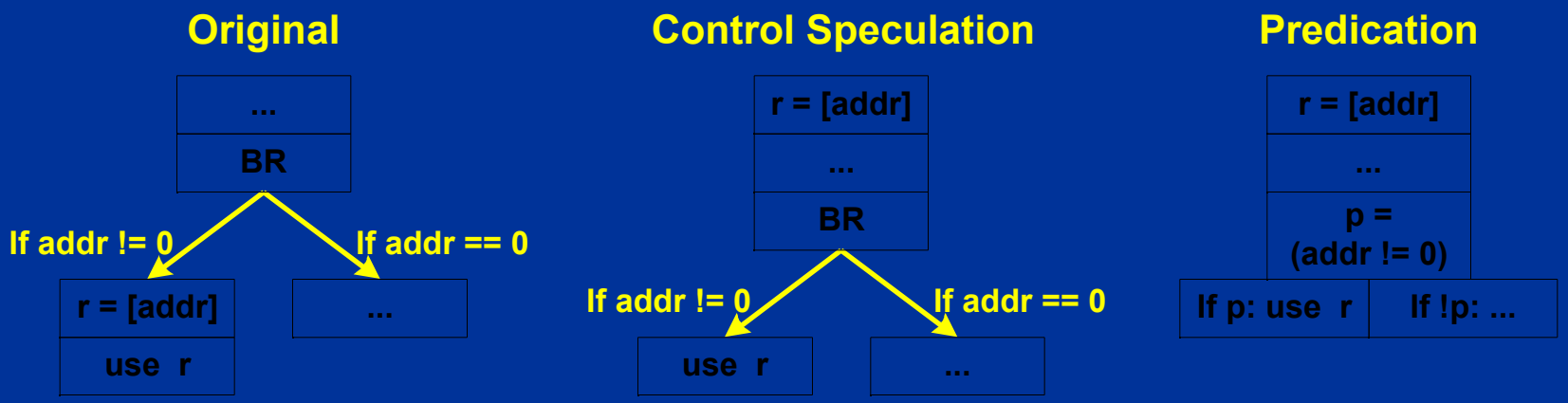
IMPACT Research Group
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
http://www.crhc.uiuc.edu/IMPACT/

Itanium Development History



Itanium Architecture Overview

- Itanium design goal: enhance scalability of parallelism by moving complex decisions to the compiler
  - Bundling: enables static scheduling by communicating instruction parallelism
  - Predication: allows compiler to optimize across multiple paths by providing an alternative to control flow
  - Speculation support: allows compiler to select specific instructions for early execution



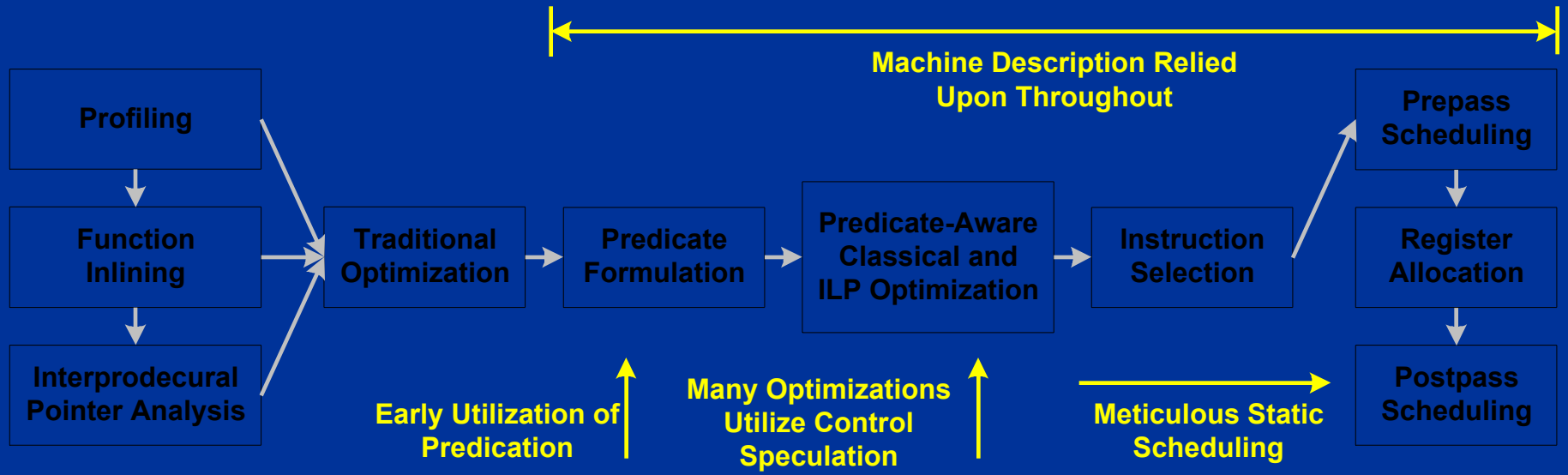
Itanium Compilation Landscape

- Increased reliance on the compiler for performance
  - Explicit control of the architecture: realities of modern microarchitecture have become visible at software level
  - Particular problems: effects of runtime uncertainty
    - Control resolution, variable memory latency, etc.
  - Solutions from EPIC/VLIW research
    - Memory disambiguation, profiling
    - Static scheduling, control speculation, predication

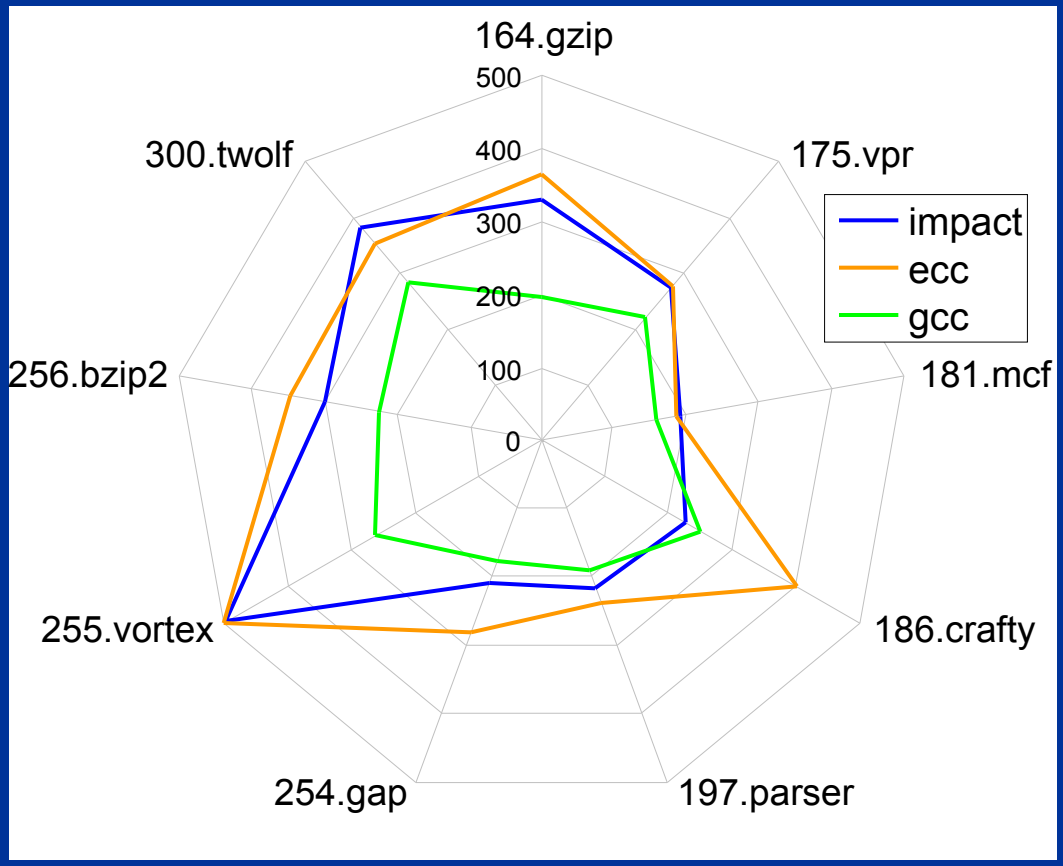
	Applicability	Public/ Proprietary	Peephole opti level	ILP opti level	Extensibility
gcc	Very High	Public	Low	Very Low	Low
ecc	High	Proprietary	High	High	High
IMPACT	Medium	Future Public	Medium	Very High	Very High

IMPACT Compiler

- IMPACT compiler supports ILP compilation and research
  - Extensible framework for easy implementation of new optimizations
  - Versatile and pervasive intermediate representation (IR)
  - Comprehensive predicate-aware dataflow and predicate analyses
  - Direct analysis from IR at most stages of compilation
  - Advanced interprocedural pointer analysis

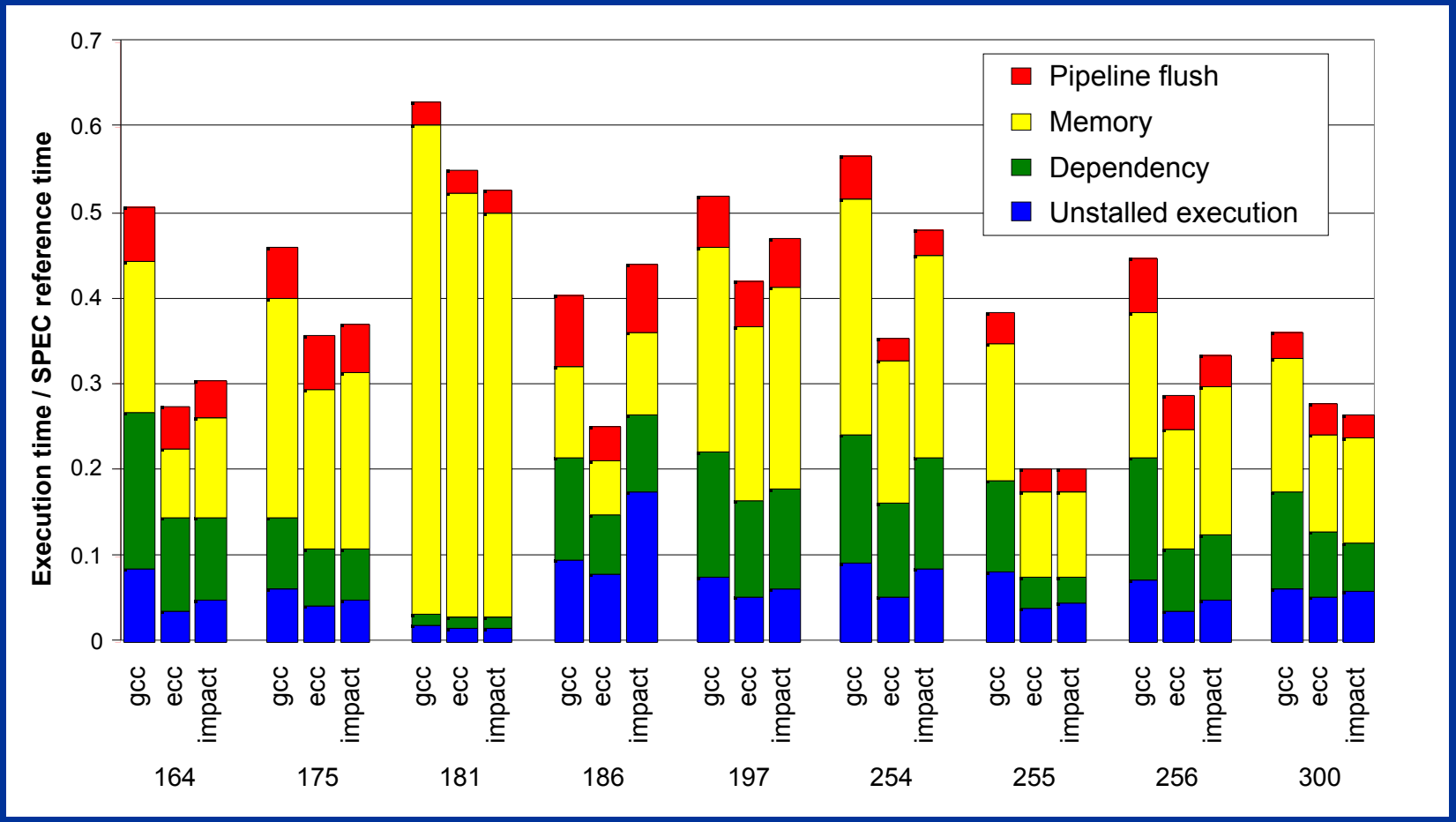


SPECcpu2000 Ratio Comparison



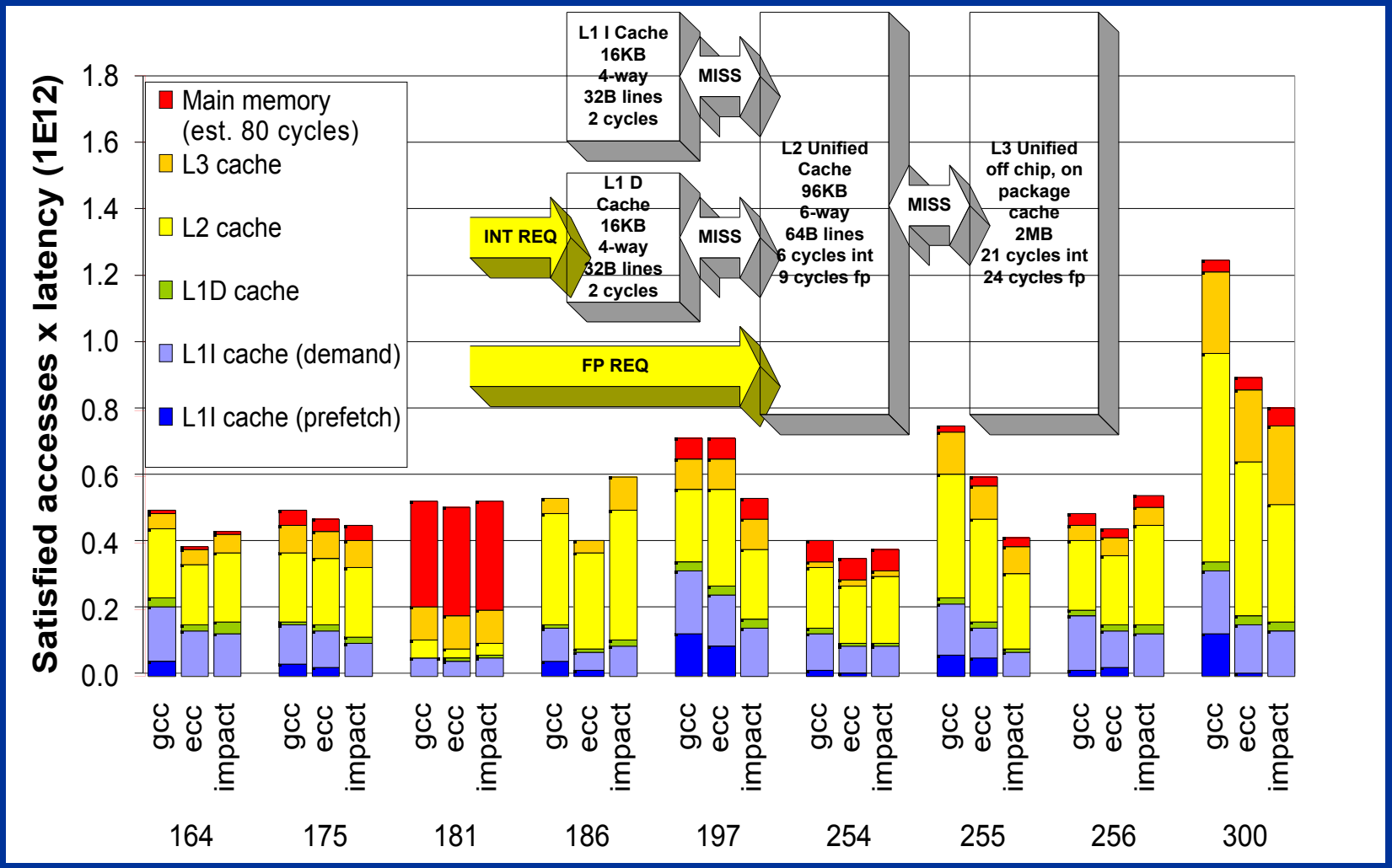
Dual processor HP i2000 Itanium 800 MHz with 2MB L3 cache
linux kernel 2.4.7 with PMC patch, gcc 2.9.6, ecc 5.0.1 beta
(ecc measurements not official Intel results)

Cycle Accounting Breakdown



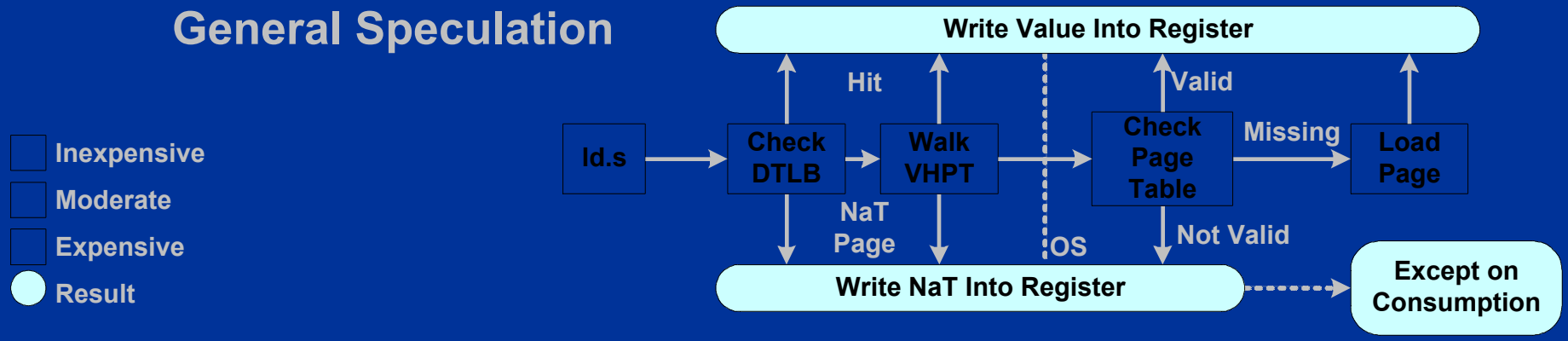
Measured machine execution cycles using performance monitoring counters

Performance of Memory Hierarchy

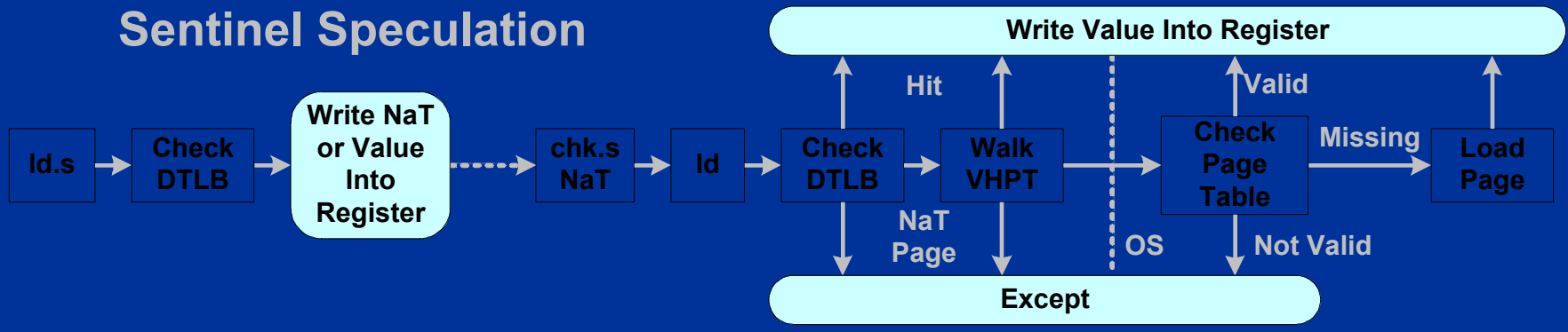


Approaches to Control Speculation

General Speculation



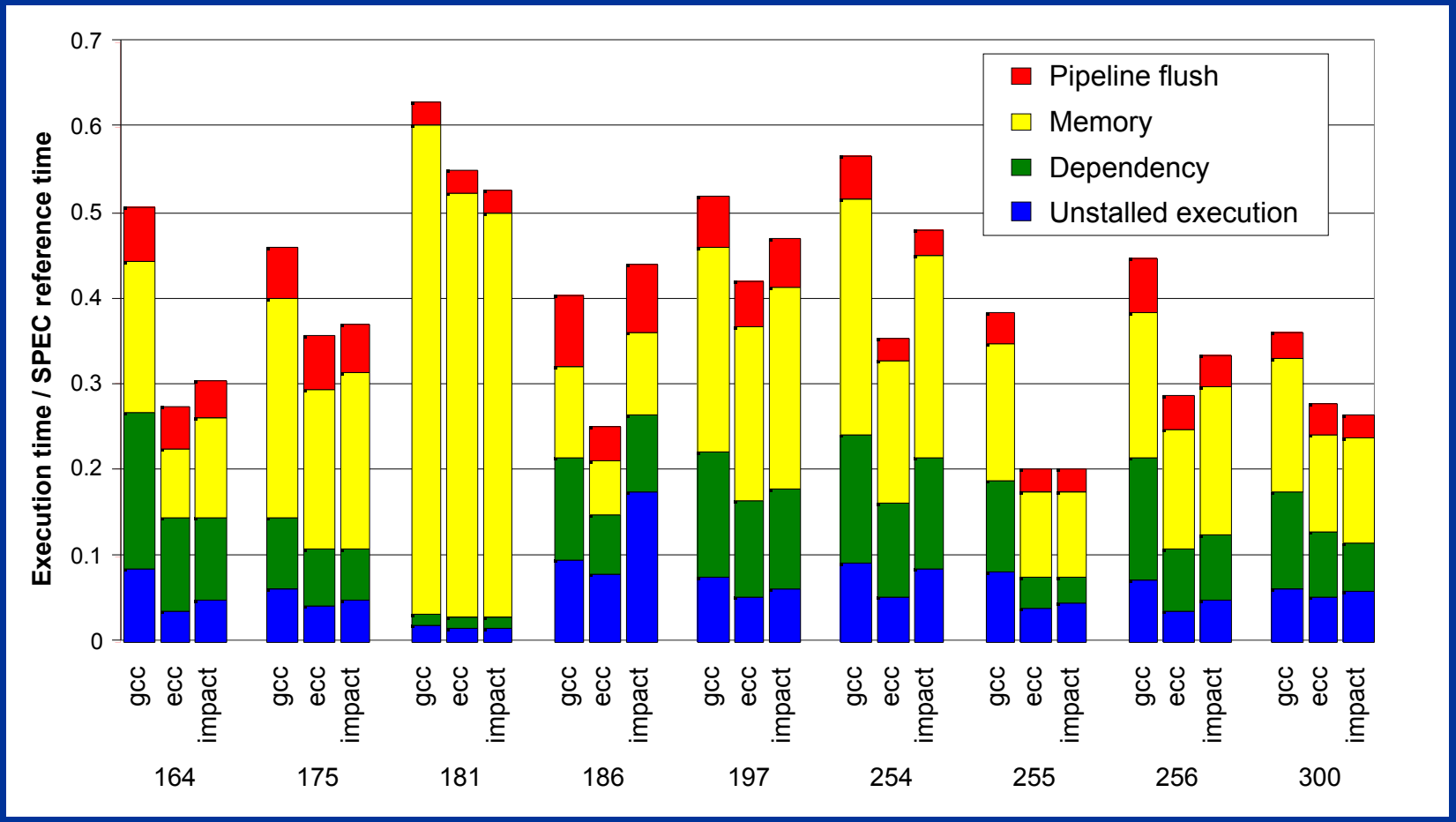
Sentinel Speculation



Architecture and OS Support

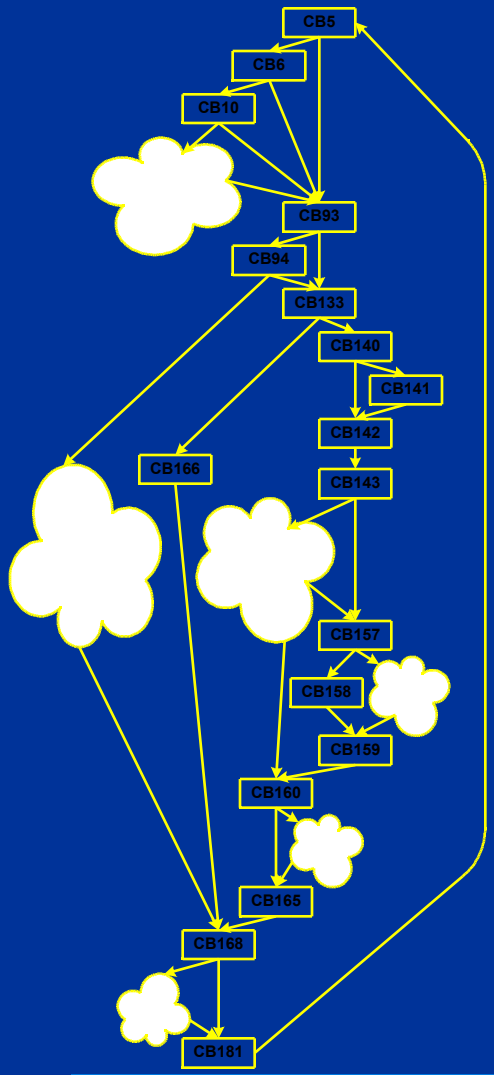
- IPF allows OS to mix and match general and sentinel speculation models
- Linux kernel modifications for general speculation
  - Basic support (system-wide general speculation): modify Default Control Register (DCR) not to defer transparent exceptions
  - Basic support caused page fault on every speculative NULL pointer access
  - Solution: install page 0 translation as a NaT page
  - Advanced support (selective general speculation): Bit in binary header indicates presence of recovery code. If no recovery code, ED bits in page table entries are cleared, overriding the DCR setting, so that no serviceable faults are deferred

Cycle Accounting Breakdown Revisited



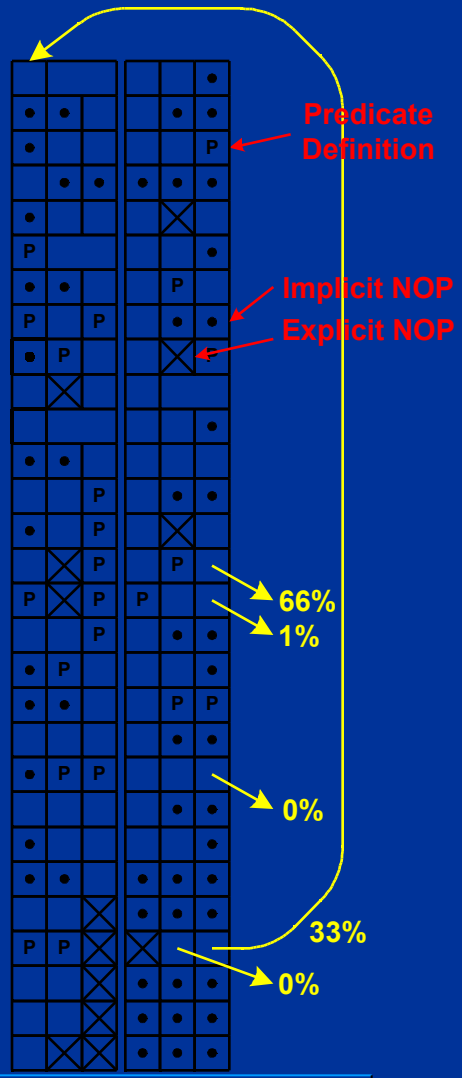
Measured machine execution cycles using performance monitoring counters

Hyperblock Transformation



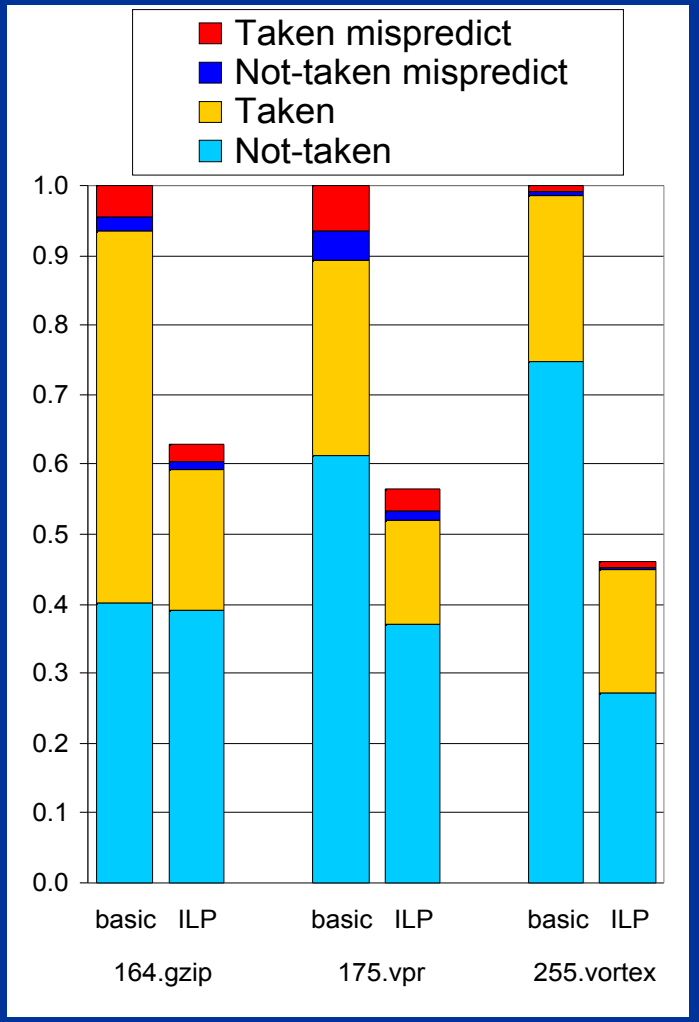
Example from *164.gzip* deflate()

- Commonly executed code pulled together into a straight-line sequence
- Serial branch execution replaced with parallel predicate evaluation
- Compromise between ILP and code size
  - Bundling reduces explicit horizontal NOPs
  - Interlocking reduces explicit vertical NOPs

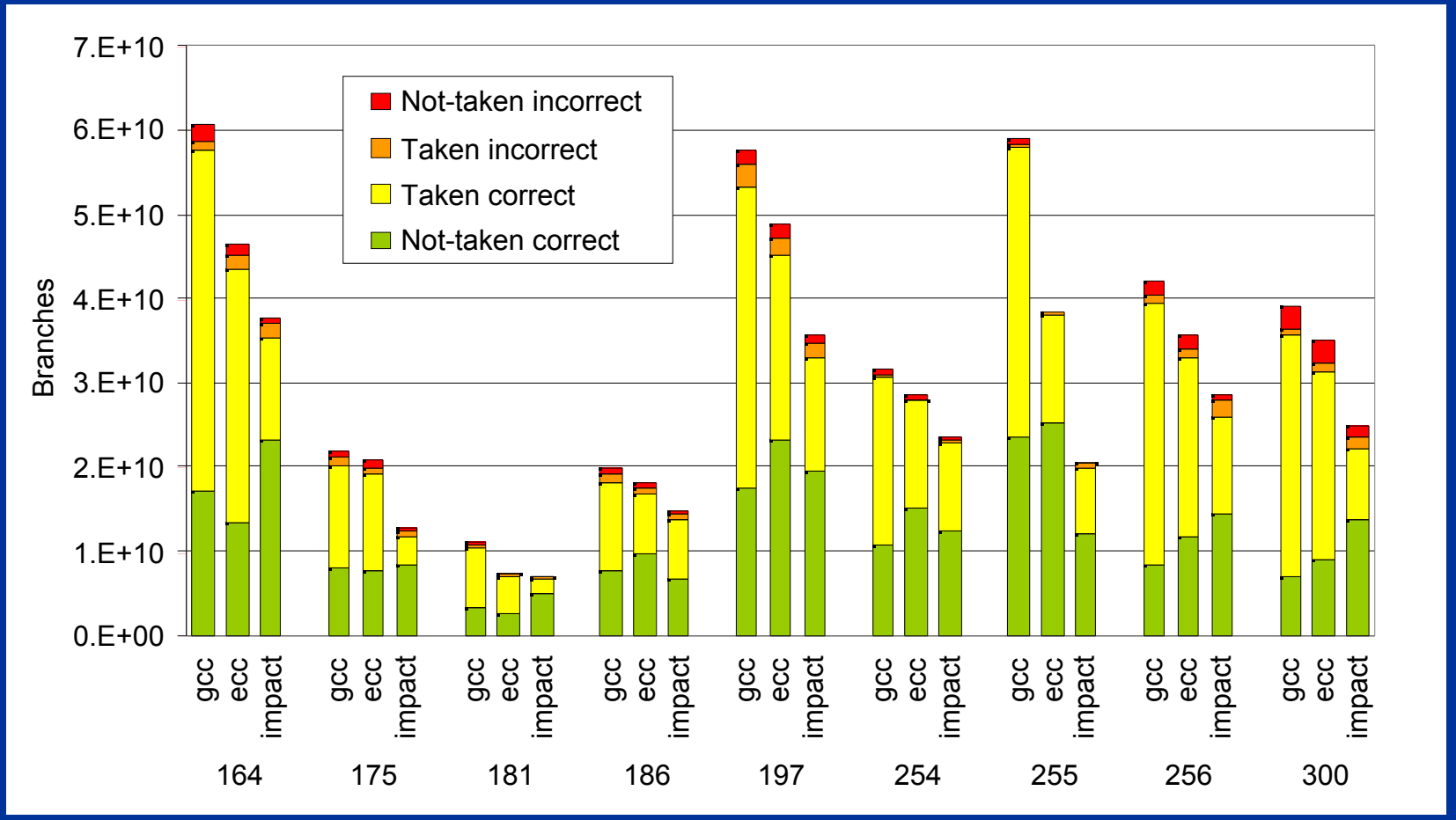


Branch Effects of ILP Transformations

- ILP compilation eliminates branches
  - Predication and branch combining
  - Loop unrolling
- Reduction in total dynamic branches
- Proportionally fewer taken branches
- Reduction in mispredicted branches



Comparison of Branch Behavior



Challenges and Future Directions

- Profile dependence and accuracy
  - Getting profile into build systems
  - Transparent runtime data collection and re-optimization, making more use of the Itanium performance monitoring counters
- Effectiveness of ILP compilation algorithms
  - New algorithms to increase both aggressiveness and stability of speculation and predication
  - Program level ILP transformations
- Memory subsystem improvements
  - More efficient pointer analysis algorithms
  - Memory data flow analysis and optimization
- More results are being derived for presentation at Microprocessor Forum

Acknowledgements

- Former IMPACT members
  - David August, Ben-Chung Cheng, Daniel Connors, Kevin Crozier, Brian Deitrich, John Gyllenhaal, Richard Hank, Teresa Johnson, Dan Lavery, Scott Mahlke, Le-Chun Wu
- Intel Itanium Team
  - Carole Dulong, John Crawford, Dan Lavery, Steve Skedzielewski, Jim Pierce
- HP Itanium Team
  - Richard Holman, Vatsa Santhanam, Carol Thompson, Richard Hank
- HP Labs PD Team
  - Bob Rau, Mike Schlansker, Vinod Kathail, Scott Mahlke
- HP Labs Linux Team
  - Brian Lynn, David Mosberger, Hans Boehm, Stephane Eranian
- HP Philanthropy - 9 i2000 Itanium machines, expedited
  - Karen Fontana, Tony Napolitan, Ralph Hyver, Chris Hsiung, Rob Bouzon
- Intel - 2 alpha/beta Itaniums
  - Carole Dulong, Richard Wirt, John Crawford